Low Complexity Turbo-like Codes

Dariush Divsalar, Sam Dolinar, Fabrizio Pollara

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA
Phone: (818) 393-5138, Fax: (818) 354-6825
E-mail: Dariush.Divsalar@jpl.nasa.gov

Abstract: *This paper*[1] *addresses the iterative decoding analysis, design, and performance of low complexity turbo like codes. For analysis, we model the density of extrinsic information in iterative turbo decoders by Gaussian density functions. We view the evolution of these density functions through the iterative decoding of turbo-like codes as a nonlinear dynamical system with feedback. Iterative decoding of turbo codes and serially concatenated codes are first analyzed based on this method for large block size. A discussion is presented for short block sizes. Then the analysis is generalized to serial concatenations of mixtures of different outer and inner constituent codes. Design examples are given to optimize mixture codes to achieve low iterative decoding thresholds on signal-to-noise ratio. Finally, based on our analysis we propose a guideline for selection of component codes appropriate for iterative decoding.*

Keywords: turbo-like codes,iterative decoding analysis, concatenated code mixtures.

1. Introduction

In this paper, we analyze turbo codes and serially concatenated codes by approximating the density functions for the extrinsics as Gaussian densities, and then computing the mean and variance in the Gaussian density evolution. This approximation was used to obtain a threshold on minimum bit signal-to-noise ratio E_b/N_0 for LDPC codes [4], based on using only the means of Gaussian densities. First we determine the input and output Gaussian means and variances of the individual SISO modules by simulation. A Similar method was also used by El Gamal [1]. For concatenated codes with two component codes such as parallel and serial turbo codes, we can plot the output SNR versus the input SNR for one component decoder, and the input SNR versus the output SNR for the other component decoder. If the two curves do not cross, then the iterative decoder converges.

[1]This work was funded by the TMOD Technology Program and performed at the Jet Propulsion Laboratory, California Institute of Technology under contract with the National Aeronautics and Space Administration.

We used all the assumptions made by Richardson and Urbanke for very large block sizes (essentially when the block size and the number of iterations go to infinity but the number of iterations is much less than roughly the log of the block size corresponding to the girth of the graph representing the overall code, then the effects of cycles on performance can be ignored). A concentration theorem [11], [6] can be used to make these results more rigorous. The concentration theorem says that the average bit error probability concentrated around the ensemble average of the bit error probability over all possible graphs representing a given code, or over all interleavers in the case of turbo codes, when the block size goes to infinity. Such convergence is exponential in the block size, and, as the block size goes to infinity, the graphs representing the code can be considered loop-free (locally tree-like). Such an assumption for turbo codes was argued in [6], based on the decay of dependencies of messages that are far apart from each other on the trellis (similar to the concept of finite-length traceback in Viterbi decoding).

2. A Model for Decoder Convergence

Consider a parallel or a serial turbo code with two constituent encoders. The turbo decoder is based on two SISO modules [2]. The iterative decoder can be viewed as a nonlinear dynamical feedback system. Extrinsic information messages are passed from one decoder to the other.

With large interleavers, the extrinsic information messages are independent and identically distributed, given, say that the all-zero codeword is transmitted (corresponding to, say, transmission of +1's on the channel). Each message is modeled by a Gaussian random variable with mean μ_i and variance σ_i^2 at the ith iteration, and the signal-to-noise ratio (SNR) of this random variable is defined as SNR $\triangleq \mu_i^2/\sigma_i^2$. If the consistency assumption is used, then $\sigma_i^2 = 2\mu_i$.

Consider the input and output SNR's for each decoder at each iteration as shown in Fig. 1. These are denoted $\mathrm{SNR1_{in}}$, $\mathrm{SNR1_{out}}$, $\mathrm{SNR2_{in}}$, $\mathrm{SNR2_{out}}$, and they represent the SNR's associated with the extrinsic information messages, not the SNR associated with the channel observations. A nonzero E_b/N_0

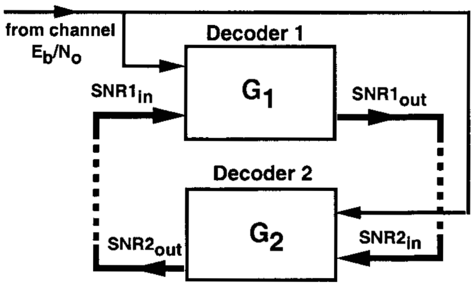

Figure 1: Analysis of turbo decoding as a nonlinear dynamical system with feedback using Gaussian density evolution.

from the channel enables decoder 1 to produce a nonzero $\mathrm{SNR1_{out}}$ for the output extrinsic information despite starting with $\mathrm{SNR1_{in}} = 0$. For a given value of E_b/N_0, the output SNR of each decoder is a nonlinear function of its input SNR, denoted by G_1 for decoder 1 and G_2 for decoder 2 as shown in Fig. 1. We have $\mathrm{SNR1_{out}} = G_1(\mathrm{SNR1_{in}}, E_b/N_0)$ and $\mathrm{SNR2_{out}} = G_2(\mathrm{SNR2_{in}}, E_b/N_0)$. Also, $\mathrm{SNR2_{in}} = \mathrm{SNR1_{out}}$, and thus $\mathrm{SNR2_{out}} = G_2(G_1(\mathrm{SNR1_{in}}, E_b/N_0), E_b/N_0)$.

We can test the decoder convergence by plotting the output SNR of decoder 1 versus its input SNR, and the input SNR of decoder 2 versus its output SNR, as shown in Fig. 2. In this figure we considered

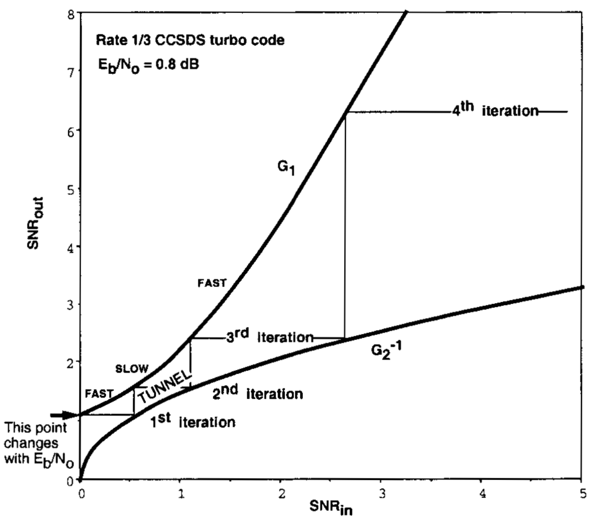

Figure 2: Iterations and convergence of a turbo decoder.

a rate 1/3 CCSDS turbo code [7] consisting of two 16-state systematic recursive convolutional codes. Encoder 1 is rate 1/2, and encoder 2 is rate 1, as its systematic bits are punctured to make the overall code rate 1/3. The upper curve corresponds to the input-output function G_1 for decoder 1, and the lower curve corresponds to G_2^{-1} for decoder 2.

Figure 2 also graphically shows the progress of the decoder's iterations. The improvement in the SNR of the extrinsic information, and the corresponding improvement in the decoder's bit error rate, follows a staircase path reflecting at right angles between the curves corresponding to G_1 and G_2^{-1}. The steps in this staircase are large when the bounding curves are far apart, and small when they are close together. Where the curves are closest together, the improvement in bit error rate slows down, as many iterations are required to bore through the narrow *iterative decoding tunnel* between the curves. If the iterative decoder successfully passes through the tunnel, convergence becomes very rapid as the two curves get farther and farther apart at higher SNRs. This means that as the block size goes to infinity the bit error rate goes to zero as the number of iterations increases.

The initial displacement of the G_1 curve for $\mathrm{SNR1_{in}} = 0$ is dependent on the E_b/N_0 due to the channel observations. If we reduce E_b/N_0 from the value of 0.8 dB used in Fig. 2, then at some point the two curves will just touch each other. That value of E_b/N_0 represents the iterative decoding threshold. The iterative decoding tunnel will be closed at the SNR where the two curves touch, and the staircase path will not go past this point. The bit error rate will settle to a nonzero value determined by this finite SNR. Conversely, if E_b/N_0 is greater than this threshold the decoder converges and the bit error rate goes to zero as the iterations increase.

3. Concatenated Codes with Mixed Inner or Outer Codes

The analytical method can also be applied to discover combinations of constituent codes whose individual strengths and weaknesses complement each other. Turbo-like concatenated codes can then be constructed using a mixture of such complementary constituent codes that outperform codes formed from either constituent alone.

An example of constituent codes suitable for such a construction is the following. A repetition-3 outer code is serially concatenated with two possible rate-1 inner codes, through an infinitely long random interleaver. One code is the two-state accumulator code (octal 1/3), and the second is a four-state code (octal 5/7). The overall code rate in each case is 1/3. When the inner code is the accumulator code, this code can be analyzed using the $\mathrm{SNR_{out}}$ versus $\mathrm{SNR_{in}}$ characteristics of the individual constituent codes, as shown in Figure 3. The SNR characteristic of the outer repetition-3 code is a straight line with slope 1/2. The SNR characteristic of the accumulator code has slope lower than 1/2 for small values of input SNR, and its slope increases very slowly with increasing input SNR. An E_b/N_0 of 0.49 dB is just sufficient to keep open a long narrow tunnel for

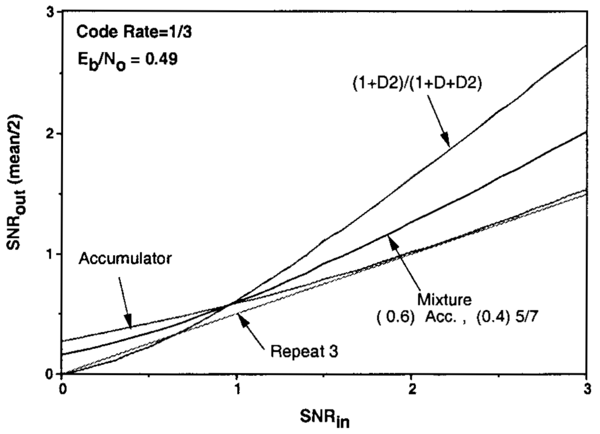

Figure 3: Example of a concatenated code with mixed inner codes.

successful iterative decoding. Thus, a great many iterations are required when E_b/N_0 is close to this code's iterative decoding threshold. In contrast, the slope of the SNR characteristic of the rate-1 four-state code is much higher than 1/2 in the SNR region where the accumulator is having the most difficulty. If this more powerful 4-state code were to replace the 2-state accumulator as the inner code, the iterative decoder would converge much faster and at lower E_b/N_0 once the inner code's SNR increased past a value of roughly 1. Unfortunately, the performance of the 4-state, octal 5/7 code falls apart completely at low values of input SNR. At the initial iteration, when the input SNR of the extrinsics is equal to zero, the output SNR for this code is also zero, because the code is rate-1 and it includes a nontrivial feedforward polynomial. In this case, the value of each input bit is dependent on an infinite number of channel symbols, and there is no redundancy in the code to help out. The result is that the information from the channel is completely useless without some additional extrinsic information to modify the *a priori* probabilities. In contrast, a decoder for the lowly accumulator code is able to start its iterations with a modest nonzero output SNR, because in this case there is no feedforward component, and each input bit is only dependent on the values of two successive symbols from the channel.

An improved concatenated code can be formed by using a mixture of the accumulator code and the octal 5/7 code as the inner code. Figure 3 also shows the SNR characteristic of a mixed inner code for which 60% of the input bits are encoded by the accumulator and 40% are encoded by the octal 5/7 code. This mixed code gets some nonzero initial output SNR from its accumulator component, and thus avoids the startup problem of the pure 5/7 inner code. At the same time, its SNR characteristic curve picks up some of the higher slope and sharper curvature of the 5/7 code's curve, thus shortening and widening the iterative decoding tunnel at E_b/N_0 = 0.49 dB, which results in faster decoder convergence and allows the iterative decoding threshold to be reduced further. Figure 4 shows the SNR characteristic curves for the same codes in Figure 3 when E_b/N_0 is reduced to 0.00 dB. From this figure we

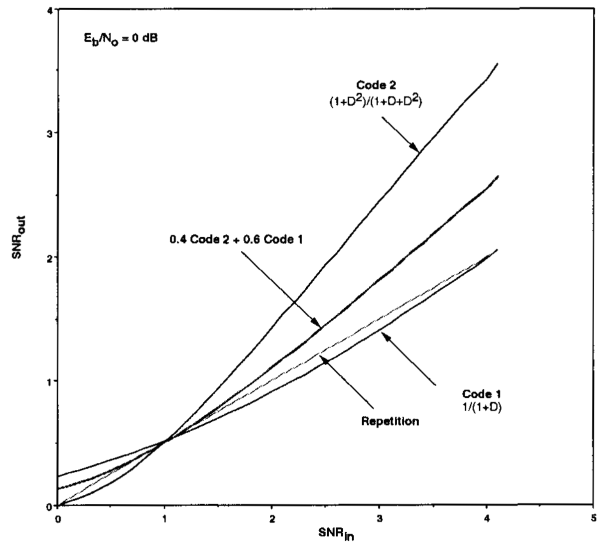

Figure 4: Reduced decoding threshold for a concatenated code with mixed inner codes.

see that $E_b/N_0 = 0$ dB is just sufficient to decode the concatenated code formed from the 60-40 mixture, but it is clearly below the iterative decoding threshold when either inner constituent code is used without the other. The optimum mixing proportion was obtained by evaluating the SNR characteristic for different mixes. In this case, almost a half-dB of improvement is obtained by mixing the two inner codes in an optimal proportion.

The analysis method for obtaining the SNR characteristic curves in the previous two figures was based on two assumptions. First, it was assumed that the probability density of the output extrinsics from both the inner and outer codes can be approximated as Gaussian at each iteration. Second, it was assumed from the consistency condition [3] that each Gaussian density is fully characterized by its mean (i.e., SNR = mean/2). For the case of mixed codes, the assumption of a pure Gaussian density at each step of the iterations seems to contravene the analysis. As similarly observed in [4] for irregular low-density parity check codes with varying degrees of connectivity between variable nodes and check nodes, what might start out as a pure Gaussian density output from each of the inner decoders becomes a mixture of two Gaussians (at different means) when the outer decoder takes outputs from two different inner decoders after passing through a random infinitely long interleaver. Then a mixture of two Gaussians at the input of the repetition-3 outer decoder becomes a mixture of three Gaussians at the output of the outer decoder or when fed to the input of the inner decoder. Thus, for the case when the outer code is a simple repetition and the inner code is a mixture, it is analytically established that a pure Gaussian density at the output of the inner decoder will imply a non-Gaussian density at the output of the outer decoder, and thus the assumption that both outputs are purely Gaussian cannot be exactly correct. However, it is empirically observed that a pure Gaussian assumption at each point in the iteration is nonetheless a robust model for determining performance thresholds.

Next we consider an example showing that it is also possible to achieve a low decoding threshold by using a mixture of outer codes concatenated with a single inner code. The inner code in this example is a 4-state rate-1 code (octal 1/7). Like the 2-state accumulator code, this code does not suffer from the extreme startup deficiency of the 4-state octal 5/7 rate-1 code, because the octal 1/7 code lacks multiple feedforward connections and each of its input bits is affected by only three channel symbols. The outer code is a mixture of two rate-1/2 codes, a simple repetition-2 code and a convolutional (5,7) code. The optimal mixing proportion in this case is to send about 2/3 of the bits to the repetition code and 1/3 of the bits to the convolutional code. Figure 5 shows the SNR characteristics of the individual codes and the optimally mixed outer code. We see

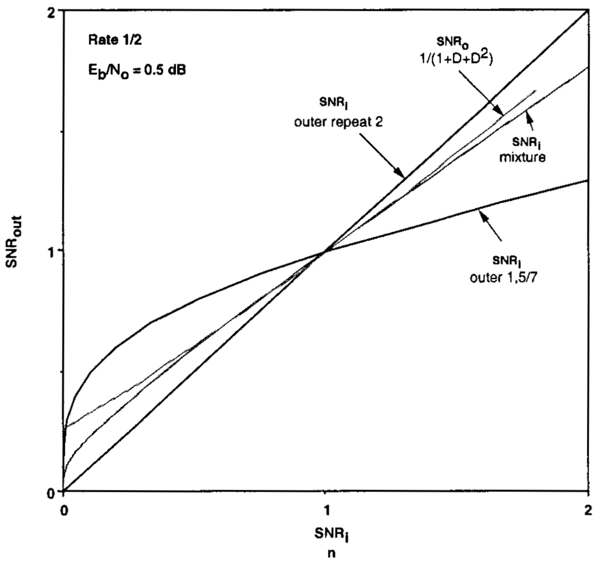

Figure 5: Decoding threshold near capacity limit for a rate-1/2 concatenated code with mixed outer codes.

that the SNR characteristics predict an iterative decoding threshold of about 0.5 dB, which is only about 0.3 dB above the capacity limit for rate-1/2 codes.

Another rate-1/2 code with even lower complexity and lower iterative decoding threshold can be constructed as follows. Start with an outer recursive convolutional code, octal (1,5/7). Send two-thirds of the parity bits (using a 100 puncturing pattern) directly to the channel. Send the remaining one-third of the parity bits and all of the systematic (information) bits through an (infinitely long) interleaver to an inner 2-state rate-1 accumulator code. Figure 6 compares the iterative decoding performance of this code with that of a more straightforward serial concatenation of the octal (1,5/7) code (unpunctured) with the 2-state accumulator code. We see that the

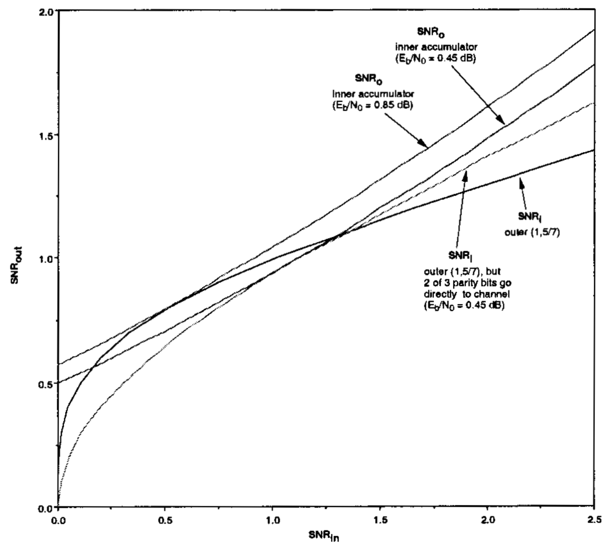

Figure 6: Decoding threshold near capacity limit for a hybrid concatenated code.

straightforward serial concatenation requires a minimum E_b/N_0 of 0.85 dB, because the sharp curvature of the SNR characteristic of the outer (1,5/7) convolutional code requires the inner accumulator code to have a fairly high starting output SNR value. The SNR characteristic of the hybrid outer code, sending two of every six bits directly to the channel, is not as sharply curved. This hurts the convergence rate at high SNR, but it widens the iterative decoding tunnel at its narrowest constriction and thus allows the accumulator's characteristic curve to drop substantially. Successful iterative decoding for the this rate-1/2 code can take place at an E_b/N_0 of about 0.45 dB, just 0.27 dB above the capacity limit.

4. Analytic Estimates of Gaussian Density Evolution for 2-state Constituent Codes

Analytic expressions for the evolution of the mean value of the extrinsics can be obtained for concatenations using 2-state constituent codes. The analysis method parallels that used in [4] for LDPC codes. Consider a linear block code, assume that the all-zero codeword is transmitted, and use the mapping $0 \to 1$, $1 \to -1$ for transmission through a binary input AWGN channel.

4.1. Analysis for LDPC codes

For iterative decoding of LDPC codes using a belief-propagation network, there are two types of nodes and three types of extrinsic information messages. There are *variable nodes* corresponding to coded symbols and *check nodes* corresponding to the parity check equations. The variable nodes send messages v to the check nodes and receive messages λ_c from the channel and u from the check nodes. The check nodes send messages u to the variable nodes and receive messages v from the variable nodes.

The message v going from a variable node to a given check node is computed as a linear sum of the incoming channel message and the extrinsic messages from the other check nodes.

$$v = \lambda_c + \sum_{m=1}^{d_v - 1} u_m \tag{1}$$

where d_v is degree of the variable node, and $\{u_m, m = 1, \ldots d_v - 1\}$ are the incoming messages from the $d_v - 1$ other check nodes connected to the given variable node. The message going from a check node to a variable node is computed as a nonlinear function of the extrinsic messages from the other variable nodes connected to this check node. For the sum-product algorithm, the result is expressed as

$$\tanh\left(\frac{u}{2}\right) = \prod_{m=1}^{d_c - 1} \tanh\left(\frac{v_m}{2}\right) \tag{2}$$

Using the Gaussian approximation and the consistency condition, we only need to compute the mean of the extrinsic messages. The mean $\bar{v}$ of the message from a variable node to a check node is simply the sum of the mean $\bar{\lambda}_c$ of the channel message and the means $\bar{u}$ of the incoming messages $\{u_m\}$ from the other check nodes,

$$\bar{v} = \bar{\lambda}_c + (d_v - 1)\bar{u} \tag{3}$$

For an AWGN channel, $\bar{\lambda}_c = \frac{2}{\sigma^2}$, where $\frac{1}{2\sigma^2} = \frac{k}{n} \frac{E_b}{N_0}$.

At the next step in the iterative process, the mean $\bar{u}$ of the message from a check node back to a variable node is computed by averaging (2) over the assumed Gaussian densities for u and $\{v_m\}$. Define $\psi(\mu)$ to be the average of $\tanh(y/2)$ over a Gaussian random variable y with mean μ and variance 2μ,

$$\psi(\mu) = \int_{-\infty}^{\infty} \tanh\left(\frac{y}{2}\right) \frac{1}{\sqrt{4\pi\mu}} \exp\left(-\frac{(y-\mu)^2}{4\mu}\right) dy \tag{4}$$

In terms of $\psi(\cdot)$ the update equation for the mean of a message from a check node to a variable node is

$$\psi(\bar{u}) = [\psi(\bar{v})]^{d_c - 1} \tag{5}$$

Here we have assumed that the variable node messages $\{v_m\}$ and the check node messages $\{u_m\}$ all have the same means, $\bar{v}$ and $\bar{u}$, respectively. These update equations for LDPC codes have been obtained in [4].

4.2. Analysis for concatenated codes with 2-state inner codes

Analytic computation of the evolving mean of the extrinsic information messages for a concatenated code with convolutional constituent codes is in general a difficult problem. Approximate solutions have not produced accurate computations of iterative decoding thresholds.

We consider two versions of the 2-state convolutional code, the rate-1 accumulator code, octal (1/3) and the rate-1/2 recursive convolutional code, octal (1,1/3). Although convolutional codes are conventionally constructed on a trellis, these 2-state codes have simple Tanner graph representations [8] without loops. This graph representation is shown in Fig. 7. Here the nodes corresponding to the parity

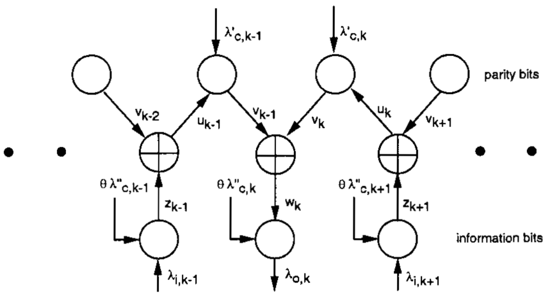

Figure 7: Tanner graph for a 2-state convolutional inner code, used to calculate extrinsic-information messages corresponding to input information bits.

bits and the information bits play the role of the variable nodes for LDPC codes, and they are connected through a set of check nodes.

We use the 2-state convolutional code, either rate-1 or rate-1/2, as the inner code of a serial concatenation or as one of the constituent codes of a parallel concatenation. The input extrinsic information corresponding to the jth information bit is denoted $\lambda_{i,j}$, and we wish to compute the output extrinsic information $\lambda_{o,k}$ for bit k. The message from the channel corresponding to the kth parity bit is denoted $\lambda'_{c,k}$ and the message from the channel corresponding to the (systematic) information bit is denoted by $\theta\lambda''_{c,k}$, where $\theta = 1$ for the case of the rate-1/2 code, and $\theta = 0$ for the rate-1 accumulator code. We also define intermediate messages $\{u_j\}$, $\{v_j\}$ between the check nodes and the variable nodes associated with the parity bits, and intermediate messages $\{w_j\}$, $\{z_j\}$ between the check nodes and the variable nodes associated with the information bits, as indicated in the figure. We can use the message passing algorithm to write the following expressions. At the variable node corresponding to the kth parity bit, the message v_k sent to the kth check node is given by $v_k = u_k + \lambda'_{c,k}$. At the kth check node, the message w_k sent to the node corresponding to the kth information bit is determined by $\tanh(w_k/2) = \tanh(v_{k-1}/2)\tanh(v_k/2)$. At the $(j+1)$th check node, the message u_j sent to the node corresponding to the jth parity bit is determined by $\tanh(u_j/2) = \tanh(v_{j+1}/2)\tanh(z_{j+1}/2)$. At the variable node corresponding to the jth information bit, the message z_j sent to the jth check node is given by $z_j = \lambda_{i,j} + \theta\lambda''_{c,j}$. At the variable node corresponding to the kth information bit, the output extrinsic information message $\lambda_{o,k}$ is computed as $\lambda_{o,k} = w_k + \theta\lambda''_{c,k}$. Using the linear equations to eliminate $\{u_j\}$, $\{w_j\}$, and $\{z_j\}$, we reduce these relationships to two nonlinear equations.

$$\tanh\left(\frac{v_k - \lambda'_{c,k}}{2}\right) = \tanh\left(\frac{v_{k+1}}{2}\right)\tanh\left(\frac{\lambda_{i,k+1} + \theta\lambda''_{c,k+1}}{2}\right) \tag{6}$$

and

$$\tanh\left(\frac{\lambda_{o,k} - \theta\lambda''_{c,k}}{2}\right) = \tanh\left(\frac{v_{k-1}}{2}\right)\tanh\left(\frac{v_k}{2}\right) \tag{7}$$

With large code blocks, we can argue that averages over the various types of messages will be independent of the bit location k. Taking the steady-state averages of these two expressions produces first an equation to be solved for the mean $\bar{v}$ of the messages $\{v_k\}$, in terms of the mean $\bar{\lambda}_i$ of the input extrinsics $\lambda_{i,k}$ and the mean $\bar{\lambda}_c = \frac{2}{\sigma^2}$ of the channel messages $\lambda'_{c,k}$, $\lambda''_{c,k}$,

$$\psi\left(\bar{v} - \bar{\lambda}_c\right) = \psi\left(\bar{v}\right)\psi\left(\bar{\lambda}_i + \theta\bar{\lambda}_c\right) \tag{8}$$

and second an equation for the mean $\bar{\lambda}_o$ of the output extrinsics $\lambda_{o,k}$,

$$\psi\left(\bar{\lambda}_o - \theta\bar{\lambda}_c\right) = \left[\psi\left(\bar{v}\right)\right]^2 \tag{9}$$

4.3. Analysis for doubly serial concatenations of 2-state codes

We can also use the 2-state convolutional code, rate-1 or rate-1/2, as an outer code or more generally as the middle code of a serial concatenation of three codes [9], [10]. Again we use a Tanner graph representation for the 2-state convolutional code. In the general case there are two types of output extrinsic messages that must be analyzed, corresponding to the information bits and to the parity bits.

almost identical to that of Fig. 7, except that the input messages from the channel $\lambda'_{c,k}$, $\lambda''_{c,k}$ are replaced by input extrinsic-information messages $\lambda'_{i,k}$, $\theta\lambda''_{i,k}$ from an inner code, corresponding to the parity bits and the (systematic) information bits (if rate-1/2), respectively. There are also input extrinsic-information messages $\lambda_{i,k}$ from an outer code, corresponding to the information bits. Now the output extrinsic messages $\lambda_{o,k}$ depend on all three types of input extrinsic messages, and the mean $\bar{\lambda}_o$ of the output messages is in general a function of the corresponding means $\bar{\lambda}_i$, $\bar{\lambda}'_i$, $\bar{\lambda}''_i$ of $\lambda_{o,k}$, $\lambda'_{i,k}$, $\lambda''_{i,k}$, respectively. To determine $\bar{\lambda}_o$, first solve the following equation for $\bar{v}$.

$$\psi(\bar{v} - \bar{\lambda}'_i) = \psi(\bar{v})\psi(\bar{\lambda}_i + \theta\bar{\lambda}''_i) \tag{10}$$

Then the solution $\bar{v}$ can be used to calculate $\bar{\lambda}_o$ as the solution of

$$\psi(\bar{\lambda}_o - \theta\bar{\lambda}''_i) = [\psi(\bar{v})]^2 \tag{11}$$

for the output extrinsics of the input bits to a middle code, and

$$\psi(\bar{\lambda}_o - \bar{\lambda}_i) = [\psi(\bar{v})]^2 \tag{12}$$

for the output extrinsics of the systematic output bits of a middle or outer code.

The graph representation for computing the extrinsic messages corresponding to the parity bits is shown in Fig. 8. Now the objective is to compute

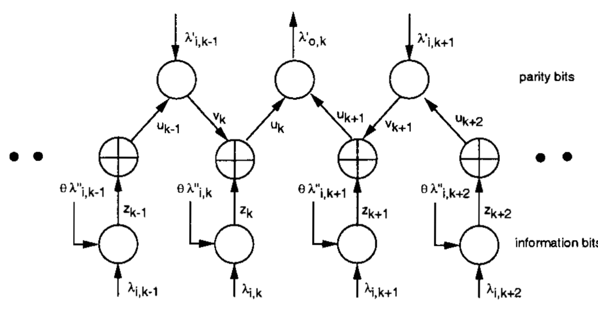

Figure 8: General Tanner graph for a 2-state convolutional "middle" code, used to calculate output extrinsic-information messages corresponding to output parity bits.

an output extrinsic-information message $\lambda'_{o,k}$ corresponding to one of the parity bits, given input extrinsic-information messages $\lambda_{i,k}$ from the input information bit and $\theta\lambda_{i,k''}$ from the output systematic bit (nonzero for rate-1/2). The message flow in this graph produces the linear equations $z_k = \lambda_{i,k} + \theta\lambda''_{i,k}$,

$v_k = u_{k-1} + \lambda'_{i,k-1}$, and $\lambda'_{o,k} = u_k + u_{k+1}$, and the nonlinear equation, $\tanh(u_k/2) = \tanh(v_k/2)\tanh(z_k/2)$. Using two of the linear equations to eliminate v_k and z_k reduces this to

$$\tanh\left(\frac{u_k}{2}\right) = \tanh\left(\frac{u_{k-1} + \lambda'_{i,k-1}}{2}\right)\tanh\left(\frac{\lambda_{i,k} + \theta\lambda''_{i,k}}{2}\right) \quad (13)$$

Again we assume a steady-state condition such that averages are independent of bit location k, and obtain the following equation to be solved for the mean $\bar{u}$ of u_k,

$$\psi(\bar{u}) = \psi(\bar{u} + \bar{\lambda}'_i)\psi(\bar{\lambda}_i + \theta\bar{\lambda}''_i) \quad (14)$$

Finally the mean $\bar{\lambda}'_o$ of the output extrinsic-information messages for the parity bits is computed in terms of $\bar{u}$ by averaging the one remaining linear equation, $\lambda'_{o,k} = u_k + u_{k+1}$, to obtain

$$\bar{\lambda}'_o = 2\bar{u} \quad (15)$$

The results in this section are for a general configuration where the 2-state code is situated amidst a series of concatenations. For the rate-1 accumulator code, set $\theta = 0$, or set $\theta = 1$ for the 2-state code with rate-1/2. If the code is used as an outer code, set $\bar{\lambda}_i = 0$. If the entire output of the rate-1/2 code is serially concatenated through an infinitely long random interleaver with an inner code, then we can also set $\bar{\lambda}'_i = \bar{\lambda}''_i$. However, we can also generally allow $\bar{\lambda}'_i \neq \bar{\lambda}''_i$ to model the case where the rate-1/2 code's systematic bits and parity bits are sent through separate interleavers to different inner codes. In particular, this applies to a code concatenation that sends the systematic bits uncoded to the channel, while permuting the parity bits and sending them to another layer of coding.

An example of this analysis method applied to a 2-state code used in a doubly serial configuration is shown in Figure 9. This is a rate-1/3 systematic code obtained by sending one copy of the information bits to the channel and two copies through a series of two rate-1 accumulators preceded by interleavers. Compared to the rate-1/3 repeat-and-accumulate (RA) code in Fig. 3, the "repeat-and-doubly-accumulate" (RDA) code in this figure has a slightly lower iterative decoding threshold of about 0.4 dB. The curves in Fig. 9 analyze this code in two pieces, with the innermost rate-1 accumulator as one constituent and the rest of the code as the second. The SNR_{out} versus SNR_{in} characteristics of these two pieces are shown along with the straight-line characteristic of the repetition-3 outer code used in the construction of the plain RA (Repeat Accumulate) code [12] in Fig. 3. We see from the figure that the SNR characteristic of the repetition-3

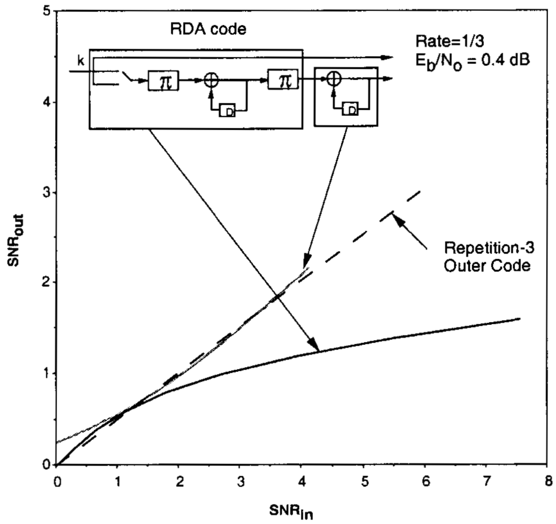

Figure 9: Iterative decoding threshold analysis for rate-1/2 LDPC codes.

outer code just barely intersects the SNR characteristic of the rate-1 accumulator code. This produces a slightly higher E_b/N_0 iterative decoding threshold for the plain RA code. More significantly, the SNR characteristic of the stronger outer code defined in Figure 9 curves sharply away from the straight-line characteristic of the repetition-3 code. This shortens the iterative decoding tunnel and enables the iterative decoder to converge much faster.

Figure 10 shows simulated performance results for the rate-1/3 RDA code compared to performance results for the rate-1/3 RA code and RDD codes.

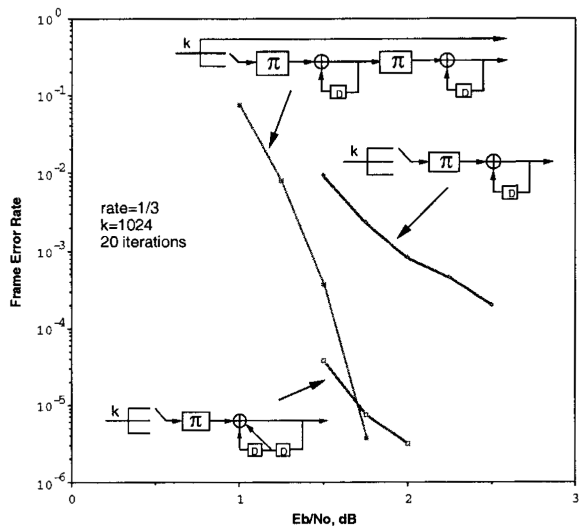

Figure 10: Simulation results for rate-1/3 RA, RDD, and RDA codes.

5. Conclusion

We modeled the density of extrinsic information in iterative turbo decoders by Gaussian density functions, and used this model to analyze the convergence of iterative decoding for turbo codes and for serially concatenated codes. Having identified the strengths and weaknesses of particular inner and outer constituent codes through their input-output SNR characteristics, we then generalized the analysis to include serial concatenations of mixtures of different outer and inner constituent codes. Such mixtures allow us to design better constituent codes that exhibit more of the strengths and fewer of the weaknesses of the individual components of the mix. The input-output SNR analysis method provides good graphical insight into understanding how to choose mixture components that complement each other. We gave examples of simple rate-1/2 and rate-1/3 mixture configurations, using component codes with at most four states, that approach their respective capacity limits within 0.3 dB to 0.5 dB.

While the general method for determining the constituent codes' input-output SNR characteristics was by Monte Carlo simulation, we also gave analytic expressions for these curves for the particular case of a 2-state constituent. These expressions cover both the rate-1, octal (1/3) accumulator code and the rate-1/2, octal (1,1/3) recursive convolutional code, used in any concatenation configuration, whether as an inner code, an outer code, or as a middle code in a series of more than two concatenated codes. To extend these analytic results to more complex constituents, the evolution of the mean of the extrinsic information can be approximately computed from stage to stage of a general code trellis, but the approximation requires ignoring a normalization factor, and to date we have not found this method to give satisfactory predictions of decoding thresholds.

REFERENCES

[1] H. El Gamal, On the theory and application of space-time and graph based codes," Ph.D. dissertation, 1999, University of Maryland at College Park.

[2] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft input soft output MAP module to decode parallel and serial concatenated codes", *The Telecommunications and Data Acquisition Progress Report 42-127, July–September 1996*, JPL, Pasadena, California, November 15, 1996.

[3] T. Richardson and R. Urbanke, "The capacity of low density parity check codes under message passing decoding, submitted to IEEE Trans. on Information Theory.

[4] S.-Y. Chung, T. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using Gaussian approximation", submitted to IEEE Transactions on Information Theory.

[5] T. Richardson, A. Shokrollahi and R. Urbanke, "Design of Provably Good Low-Density Parity Check Codes", submitted to IEEE Transactions on Information Theory.

[6] T. Richardson and R. Urbanke, "Analysis and Design of Iterative Decoding Systems", 1999 IMA Summer Program: Codes Systems and Graphical Models, Minnesota, USA, August 2-6, 1999

[7] CCSDS (Consultative Committee for Space Data Systems), "Telemetry Channel Coding", Vol. 101.0-B-4. Blue Book. Issue 4. May 1999. Available at `http://www.ccsds.org/documents/pdf/CCSDS-101.0-B-4.pdf`

[8] R.M. Tanner, "A recursive approach to low complexity codes", IEEE Trans. Information Theory, pp. 533-547, 1981.

[9] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Analysis, Design, and Iterative Decoding of Double Serially Concatenated Codes with Interleavers," IEEE Journal on Selected Areas in Communications, vol. 16, no. 2, pp. 231-244, February 1998.

[10] H. Pfister and P.H. Siegel, "On the Serial Concatenation of Rate-One Codes Through Uniform Random Interleavers," Proceedings 37th Allerton Conference on Communication, Control and Computing, Monticello, Illinois, September 22-24, 1999.

[11] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Analysis of low density codes and improved designs using irregular graphs", proceedings of the 30th Annual ACM Symposium on Theory of Computing, 1998. pp. 249-258.

[12] D. Divsalar, H. Jin, and R. J. McEliece," Coding Theorems for 'Turbo-Like' Codes," 1998 Allerton Conference, September 23-25, 1998.